Assignment – your best software project 20.02.17

1. Learning

1a) What vehicles will you and your team use to make learning a regular part of your agile iterations?

- Enhancing collaboration is the clue to learning. First of all our agile team will be multi disciplinary. People with different background bring in diverse opinions and solutions, which will lead to a better product.
- An open failure culture is very important, too. I am going to encourage the team to deal with errors openly by just doing it myself. During the daily stand ups I will always share a short story about a failure, and how I learned from it.
- Fail fast is the goal. Testing and close collaboration within team members realised by cross reviewing collected data and code will help to find errors as quick as possible. Information on the failure and solution will always be shared in the meetings.
- A close contact to the customer will help to learn as much as possible about the actual needs and values. All team members should have direct conversations with potential users to create valid narratives and a broad understanding for the problems, that should be solved.

1b) What metrics will you use to measure outcomes?

It is very important to check the actual usage of new features. We are going to use the 0,30,90 days approach combined with the following questions:

- 0 days: Is the feature usable?
- 30 days: Is the customer still using it on a regularly basis?
- 90 days: Does the feature really fulfil the customer need, that we expected to serve?

These questions should be answered with “yes” as often as possible. The percentage rate of “yes” will be monitored. It should raise over time using agile. If not, the practices and processes have to be reviewed.

1c) Who will take the lead on designing that learning? Who will take the lead on sharing it with the team in a way that drives strong, interdisciplinary discussions and links to the job of Deciding?

We are going to have an agile coach, who will be responsible for the whole organisation of the agile processes and practices. He/she will be responsible for a good agile working environment. The coach will take the lead during the planning meetings and in the group discussions. The more the agile practices are established, the less will this role be necessary.

1d) How often and in what meetings/workshops will you deliver the above?

- Daily stand up meetings every morning
- Sprint planning meetings at the beginning of each iteration
- Sprint review meetings at the end of each iteration
- Quarterly general agile practices review meetings

1e) How will you present your user stories and related items (storyboards, story maps, etc.) to drive the best possible discussions in your sprint planning meetings?

On the central wall of the agile team workspace will be a user story map, that shows the actual epics in storyboards and the corresponding user stories, sorted by priority.

1f) What about after the sprint starts? How will you work to tilt the working environment towards thinking about what makes sense vs. just creating output - those 'blue button' moments?

The open failure culture will help to improve the conversation between the team members. The product owner should be around as much as possible to be available for discussions. Changes in the plan will be handled as normal, not as a problem. During the daily stand ups the team will be motivated to articulate doubt.

1g) What are the challenges and focal points you see for linking to the job of Deciding and creating a culture of experimentation?

- It is important to not just collect information, but using it to improve the processes / Deciding. Data, which is not used is worthless.
- The best way to establish a culture of experimentation, is, to be honest about failures and processes that did not work out. If the team leader is able to say openly: "I really thought this was a good idea, but I was wrong. So how can we do better?" He/she encourages the whole team to live the culture of experimentation.

2. Deciding

2a) How long will your iterations be – and why? What might be the impact of longer iterations? Shorter?

At the beginning, the iterations will one week, which is short. Meetings on planning and reviewing will be very often. This helps to better understand the agile practices and to eliminate obstacles and problems, that will arise. Questions can be answered quickly and the collaborative culture can be trained intensively.

The longer the team works in the agile environment, the longer the sprints can be as there wont be that much questions on the actual work flow any more. I would raise the sprint length up to three weeks, depending on the capacity of the team and the jobs that should be done during the iteration.

2b) How will you groom the product backlog to increase the quality of your inputs and prioritise tasks? What inputs and what people will be most important to your pre-sprint backlog grooming sessions? Why?

- The user stories in the product backlog should be as much as evidence based as possible, meaning based on user interviews/feedback. Ideas and assumptions from the team will also be included, but marked differently.
- Every user story gets a priority, which is checked regularly. The user stories get combined in an epic to a minimum value product, to ensure the creation of working features. Details can be delivered over time or during future iterations.
- The product backlog will frequently be discussed with different team members and stake holders. The feedback from sales and service employees can be as valuable as the input from designers and developers.

2c) What is most critical to manage flow across design, development, and testing? For example, how will you make sure testing isn't backloaded to the very end of the sprint?

Testing as a part of the daily work helps to keep the work flow smooth. The code should be tested in small pieces like unit testing. In addition the testing process should be automated as much as possible. This is can only be reached in a process, but it's worth to invest continuously in this process.

2d) Which practices from XP, scrum, and kanban will you use to make the job of deciding more effective?

- We are going to use a kanban board like trello to manage the actual work flow. WIP limits will be included to ensure the right amount of workload in each section.
- Big changes during the iteration will not be allowed, like in Scrum. A new sprint will be planned and executed then.
- We are going to make sure that there is always enough slack, like mentioned in XP. Unexpected situations are normal and there should always be enough time, to react on those without screwing up the whole sprint plan.

2e) How will you evaluate the quality of your decisions and think about how to improve them?

- Frequent releases will be the prove of quality. Our plan is, to do a release every three weeks. The regular time frame is more important than the size of each release. If releasing is easy and a permanent part of the work flow, the decisions made are good. Otherwise the processes have to be reviewed.

3. Building

3a) How will you support the development and testing team? If it's a new team, which practices (from XP, scrum, and kanban) will you discuss with the team? Why those?

As we have to build an agile team for the first time, the processes need to be set up. These techniques should be used in the beginning:
- A kanban board including WIP limits to ensure a smooth work flow
- Short iteration times like recommended in XP
- Set focus on automation of integration and testing. Until the processes have been established and improved over a while, two team members will only work on the creation of automated testing and integration tools. Like this releasing code should become easily and frequently.

3b) How will you frame the 'win' on the practices you suggest for developers? Testers?

- The code should have a low defect rate
- The code should stay simple – integration should always be easy

3c) How will you discuss the best way to run testing and decide who on the team will do what?

- As we have two people in the team responsible for the test design, it is their responsibility to come up with ideas first. These will be discussed openly with the whole team to improve them and collect more valid input.

3d) How would you like to evolve the process of testing and deployment over time? What do you see as the first few steps?

Testing and deployment should become as natural as possible. The two topics should merge more and more one buy enlarging the collaboration between the team members. Short building and testing periods like unit testing will help to improve here.

3e) How will you create slack to allow the team to avoid or reduce technical debt?

During every building sprint there has to be enough spare time / slack to check the code within the team. This is as important as finishing the main user stories.

4. Managing

4a) What are the top three things you can do in your role to foster and contribute to a self-organising team?

- Decisions like importance of user stories or which agile practise to use will be made by the team, not by one person.
- Enhance collaboration by organising review processes in teams for code, user interviews etc.
- Let the team decide, when a feature is ready for release

4b) What is the role of the retrospective for you? What agenda will you use? How will you tie the results of the retrospective back to the job of learning?

Retrospectives are very important. They give valuable input on the quality of work done, the processes used and ideas for future estimations on workload. The retrospectives deliver directly the input for “learning”. Everything reported can be used to improve future processes. Important questions for the agenda could be:

- How did the release go? Did we achieve everything we wanted? If no, why?
- What obstacles came across?
- What was the best / worst thing during the last iteration?
- We used this … agile technique for the first time. How did that work out for you? Should we keep it/ improve it/ drop it?